# Indoor Localisation with Sensor fusion of PDR and single RTT Wi-FI

Annie Marandi CS19M009
Sai Rohitth Chiluka ED18B027

# Problem definition

- Hands-on approach to indoor localization using sensor fusion of PDR and Wi-Fi RTT.
- We are only considering Single access point for Wi-Fi RTT.
- Initial position is also known .

# Background

- IMU (Inertial Measurement Unit): Its a type of sensor that measures angular rate, force and magnetic field.

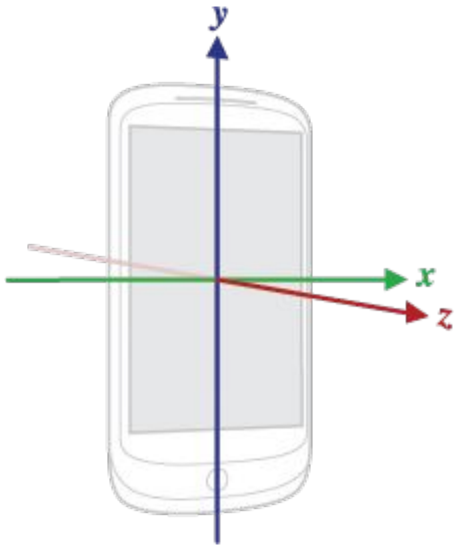
- WIFI-RTT: feature added to IEEE 802.11mc. Allow devices to measure distance to nearby Wi-Fi routers using round-trip time for signal to travel.

# IMU Types:

➢ Accelerometer: Measure acceleration

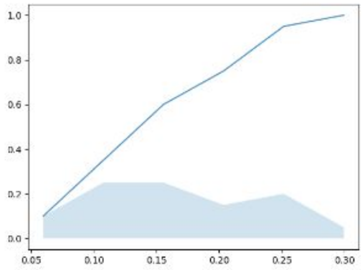
➢ Gyroscope: Angular velocity (pitch, roll, yaw)

➢ Magnetometer: measure magnetic fields



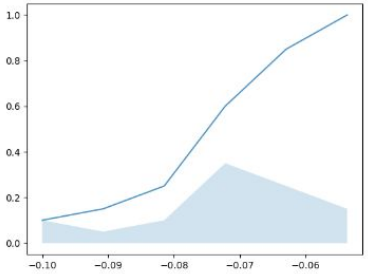
# Errors

- ➢ Bias: constant offset of the output value from input value. Eg- bias over temperature, bias instability, etc.
- ➢ Gyro-drift: from integrating angular velocity to get orientation.




Errors in each step length — Almost Gaussian

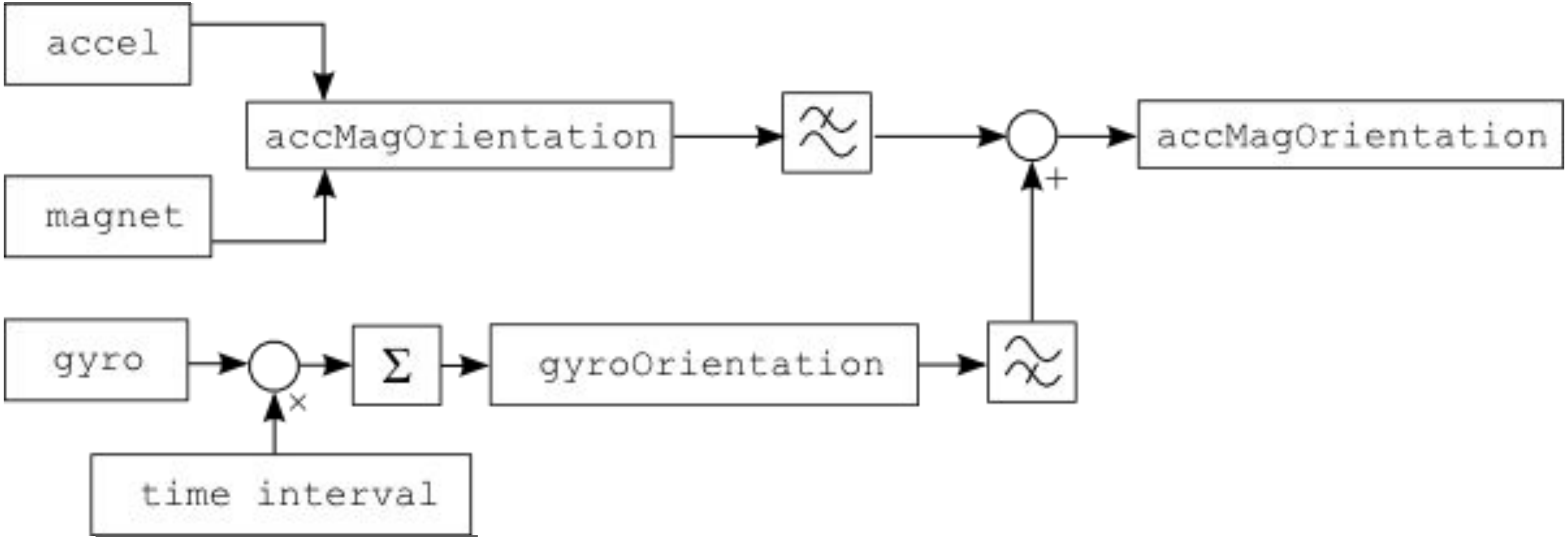
Step Length




Almost Gaussian

Orientation

# Sensor fusion using complementary filter



# Acceleration

Remove gravity component with low-pass filter:

acceleration = acceleration - (alpha * gravity + (1 - alpha) * acceleration)

Low-pass filter: Passes signals with frequency lower than the cutoff.

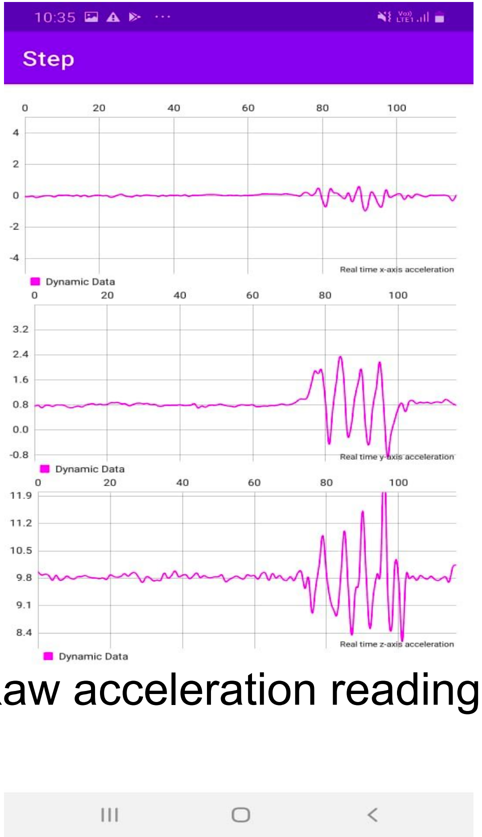
$$\alpha = dt / (RC + dt)$$

Where, $\alpha$ is the alpha, $dt$ is the time interval

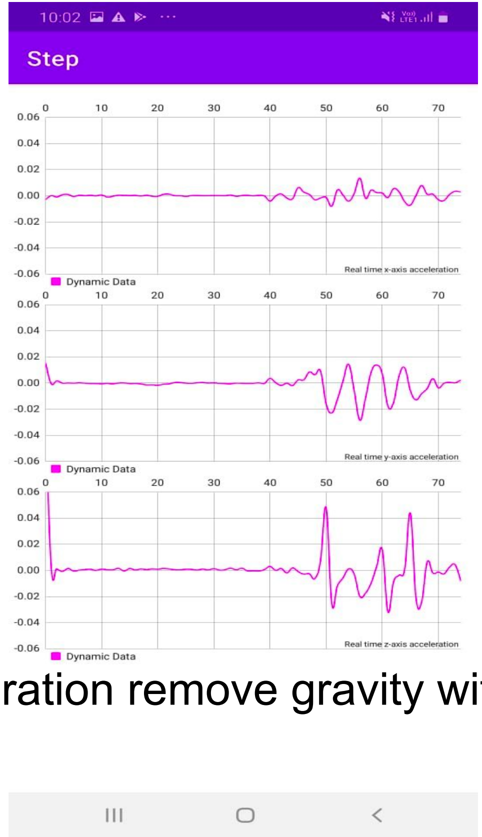
$$RC = 1/2\pi f$$

Where, $f$ is cutoff frequency

# Step detect




Raw acceleration readings




Acceleration remove gravity with LPF

# Step length

$$L = K\sqrt[4]{a_{max} - a_{min}}$$

Where, K =  0.68 - 0.37 * v + 0.15 *pow(v,2)

and, v is the average step velocity

# PDR

$$X_t = X_{t-1} + L_t \begin{bmatrix} sin(\theta_t) \\ cos(\theta t) \end{bmatrix}$$

Where, L is the step length

and, $\theta$ is the heading w.r.t phone

# RTT range simulation

- Effect of Position of AP
- Effect of Radius of circle
- Effect of update frequency

# Single RTT Localization

Using PDR estimate position and find the closest point to this estimate:

$$x = x_a + (x_p - x_a) * r/d$$

$$y = y_a + (y_{pdr} - y_a) * r/d$$

# Kalman filter

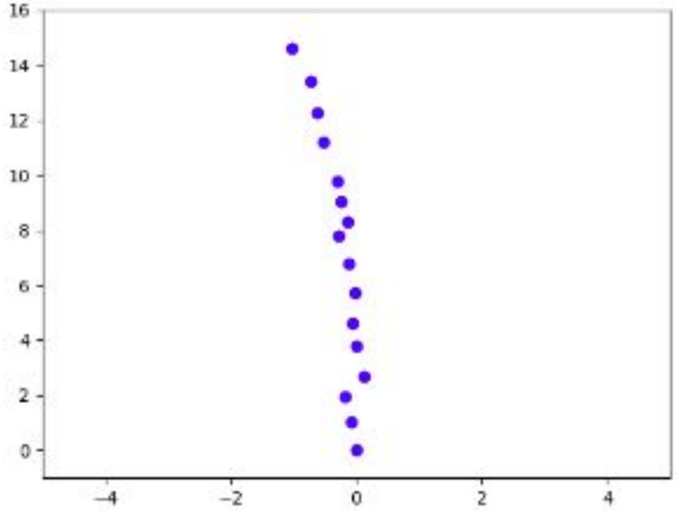
**Algorithm Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):**

$$\bar{\mu}_t = A_t\, \mu_{t-1} + B_t\, u_t$$
$$\bar{\Sigma}_t = A_t\, \Sigma_{t-1}\, A_t^T + R_t$$

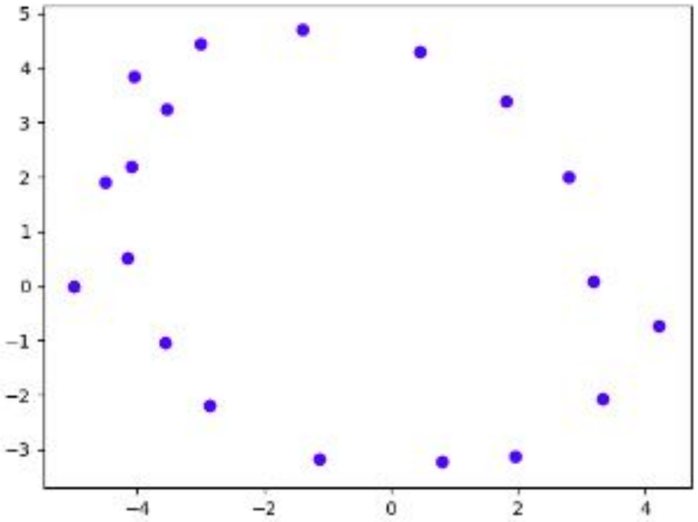
$$K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\, \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$$

**return** $\mu_t, \Sigma_t$

# Walking Simulation

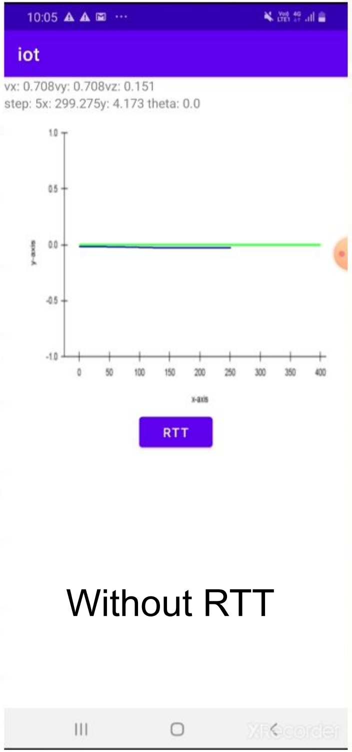



Line Simulation

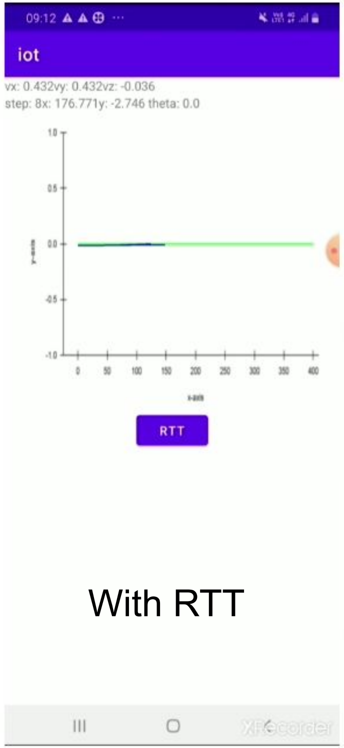



Circle Simulation

# Android App




Without RTT




With RTT

# Thank You!